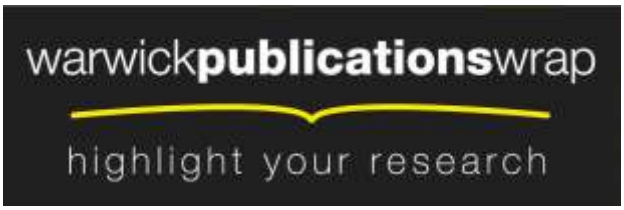

The University of Warwick

THEORY OF COMPUTATION REPORT

No. 3

FINITENESS IS MU-INEFFABLE

DAVID PARK

Department of Computer Science
University of Warwick
COVENTRY CV4 7AL
ENGLAND.

July 1974

FINITENESS IS MU-INEFFABLE

By DAVID PARK

1. Introduction

The "mu-calculus" is a formal system which arises fairly naturally when one tries to extend predicate logic to formalise arguments about programs. First-order predicate logic by itself is not adequate to express interesting assertions about programs naturally, even for the most limited sorts of programming language, as ~~was first~~ can be seen from the unsolvability results of Luckham, Park & Paterson [6]. On one view, what ~~was~~ is needed to express such things, ~~it turned out,~~ ~~was~~ is the "minimal fixpoint operator" μ. By adding such an operator to the classical primitives of predicate logic, it ~~did~~ becomes possible to formalise many arguments about programs in a nice way, and to see many previously enunciated principles for reasoning about programs as relying on the peculiar features of this operator (see, for example Park [7], de Bakker & Scott [2]).

The calculus that ar~~ose~~ises this way ~~was~~ is not obviously "classical", in that its expressive power (what it ~~could~~ can say about mathematical structures) ~~did~~ does not ~~obviously~~ appear to coincide with that of any previously studied logical formalism. Clearly, the mu-calculus ~~was~~ is strictly more expressive than first-order predicate logic (it ~~could~~ can, for one thing, express equivalences between arbitrary program schemas, so that its valid sentences ~~did~~ do not form an r.e. set, from [6]). On the other hand the mu-calculus ~~was~~ is no more expressive than classical second-order systems (any sentence of the mu-calculus ~~was~~ is translatable into an equivalent sentence of second-order predicate logic, in the style of [7]).

This paper will make rigorous the intuition that the mu-calculus is indeed strictly intermediate in expressive power between first- and second-order logics. That this is so will be seen to arise from its inability to formalise the property that its domain is finite. This is a property which can be expressed in any of the usual formulations of second-order logic, as well as in the more recently studied "infinitary" logic $L_{\omega_1\omega}$ (first-order predicate logic, but permitting certain expressions of infinite length).

On the other hand, we will see in passing that the property of being "well-founded" or "well-ordered" by a given binary relation is expressible in the mu-calculus, but not in $L_{\omega_1\omega}$, and from this that $L_{\omega_1\omega}$ and the mu-calculus are incomparable in expressive power.

The important subformalism which we call the "continuous" mu-calculus is contained in $L_{\omega_1\omega}$. This formalism appears to be adequate for expressing most facts of interest about deterministic program schemas. Nevertheless certain facts about non-deterministic schemas, e.g. that all possible execution sequences terminate, are not necessarily expressible in it (they are directly related to well-foundedness properties); ~~moreover,~~ and termination properties of deterministic schemas can profitably be looked at in this way (see [3]), and can be established using arguments formalised in the non-continuous formalism.

Rather than referring back to other formulations of the mu-calculus, we have presented a slightly different (and perhaps more easily absorbable) formulation in §2, which we hope will make this paper comparatively self-contained.

2. The Mu-calculus:

Informally, we obtain a mu-calculus by adding to first-order predicate logic a recursion operator μ, which augments the logic sufficiently to express properties of the relations computed by flow-chart schemas [6] or recursion-equation schemas [7] on arbitrary first-order structures (or data types, if you like). The μ operator is in a sense an alternative quantifier for relations, replacing the "classical" quantifiers $\forall, \exists$ on relations, but not on individuals.

Formally, there is a variety of[1] ways in which to present this calculus. Perhaps the more elegant formulation is the one made in terms of the polyadic relational systems of Park & Hitchcock [3], or de Bakker & de Roever [1], in which individual variables are suppressed, and the role of existential quantification on individuals is taken over by the composition operator between relations, which is, arguably, more natural when talking about programs. The formulation below has been chosen for the sake of comprehensibility, being based on the more traditional formalism of predicate logic.

Syntax: this is to be described in terms of an alphabet which includes the following infinite classes of symbols:

$x_0\ x_1\ x_2\ \ldots\ldots\ldots$ (individual variables)

$X_0^n\ X_1^n\ X_2^n\ \ldots\ldots\ldots$ (n-ary predicate letters), $n \geq 0$.

We will be interested in "well-formed formulas" (wffs) and "n-ary relation terms" (n-rts) defined as follows.

wffs: a class of strings containing

(I) true false $[y_1 = y_2]$

(II) $\neg Q$ $[Q \wedge R]$ $[Q \vee R]$ $(\exists y_1)Q$

for any wffs Q,R.

(III) $F(y_1, y_2, \ldots\ldots, y_n)$ for any n-rt F, $n \geqslant 0$.

n-rts: a class of strings containing

(IV) X_i^n , $i \geqslant 0$

(V) $[\lambda y_1\ y_2\ y_3 \ldots\ldots y_n . P]$ for any wff P.

(VI) $[\mu X_i^n . F]$, for any n-ary predicate letter X_i^n, and any n-ary relation term F which is formally monotone in X_i^n (see below).

($y_1, y_2, \ldots$ are used in (I)-(VI) and below to denote arbitrary individual variables.)

The classes of wffs, n-rts are ~~then~~ the smallest classes of strings ~~such that~~ satisfying (I)-(III) ~~describe~~ wffs and (IV)-(VI) ~~describe n-rts~~.

formal monotonicity: the qualification of (VI) amounts to the following - Definition: F is formally monotone in X_i^n iff every occurrence of X_i^n in F is in an even number of distinct negated sub-wffs $\neg P$ of F.

e.g. $[\neg(\exists x_1)\ [\neg X_1^2 (x_1, x_2) \vee X_2^1 (x_3)] \wedge X_1^2 (x_2, x_3)]$

is formally monotone in X_1^2, but not in X_2^1.

Substitutions: we use a slash notation, for example

$[F/X, y/x]\ P$

for substitutions for individual variables and predicate letters within wffs or rts. As usual, a strict definition requires care;

P here might need alphabetic changes of bound variables before the substitutions are made. Note that both individual variables and predicate letters may be bound, the former by $\exists,\lambda$, the latter by μ. We omit rigorous definitions of these notions here, which should be routine to the reader familiar with predicate logic.

Abbreviations: $\forall$, $\rightarrow$, $<->$, and other predicate calculus symbols will be used as abbreviations in the standard way. Note, nevertheless, the choice of $\{\exists,\wedge,\vee,\neg\}$ as basic operations, since this is of consequence to the definitions of formal monotonicity, and especially of formal continuity below.

Semantics: we hope the reader will be happy with a brief informal account. For the most part, the semantic notions should be clear from the standard semantics of predicate logic (indeed, the subsystem generated by (I)-(V) is just one version of first-order predicate logic with identity). To provide a denotation for a wff or n-rt we need a base set (a "domain"), and an assignment of relations and individuals over that set to the free predicate letters and free individual variables of the wff or n-rt. The semantics should then provide us with a truth value or relation, respectively, proceeding to do so by induction on the length of the string involved. The rules corresponding to cases (I)-(IV) are standard. "λ" is intended, of course, as an abstraction operator -- $[\lambda\, y_1y_2\ldots y_n.P\,]$, under the given assignment, denotes the relation which holds of an n-tuple $< c_1,c_2,c_3,\ldots c_n>$ just when P is true under the modified assignment which assigns c_i to y_i , $1 \leqslant i \leqslant n$. "$\mu$" is the recursion operator. F is to be regarded as a functional on relations X_i^n , which will be monotone if F is formally monotone in X_i^n . $[\mu X_i^n.F\,]$ then denotes the least fixpoint of this functional, i.e. the intersections of all relations which, when assigned to X_i^n , make X_i^n,F equivalent. Since the interpreted F is monotone as a functional of these relations, such a minimal fixpoint exists, from the Knaster-Tarski theorem (see [7]).

(Note: from a theorem of Scott and Bekic, multiple (simultaneous) fixpoints can be defined in terms of the simple notion defined here: see [3]).

Semantic Entailment, Equivalence: we use the notation

$$P \models Q$$

for semantic entailment. Thus $P \models Q$ iff every assignment which gives P the value true also satisfies Q in the same sense. Then

$$P \equiv Q \iff P \models Q \quad \text{and} \quad Q \models P$$

defines equivalence between wffs. We can extend these notions to n-rts, by saying that $F \models G$, say, iff $F(y_1 \ldots y_n) \models G(y_1 \ldots y_n)$, for all variables $y_1 \ldots y_n$; or, equivalently, iff $F(y_1, y_2 \ldots y_n) \models G(y_1, y_2 \ldots y_n)$ for all sets $\{y_1, y_2 \ldots y_n\}$ of distinct variables none of which occur free in F or G.

Examples: (taking minor cosmetic liberties with names of variables) given predicate letters Z,W intended to denote relations on the natural numbers as follows:

$Z(x)$ ----- "x=0"

$W(x,y)$ ----- "y=x+1"

the induction axiom for arithmetic can then be written as:

$$(\forall x)\ [\mu X.\ [\lambda x.\ [Z(x) \vee (\exists y)\ [W(y,x) \wedge X(y)]]]]\ (x)$$

We can also obtain a 3-rt for the addition function "z=x+y" :

$$\mu X.\ [\lambda xyz.\ [[\, Z(x) \wedge [\, y=z\ ]] \vee (\exists u)\ [W(u,x) \wedge (\exists v)\ [X(u,y,z) \wedge W(v,z)]]]]$$

and so on.

Note that the natural numbers with the intended Z,W have a categorical theory in the mu-calculus, i.e. they constitute the only model of the sentences which are true of them; the classical model-theoretic results (Löwenheim-Skolem theorems etc.) now cease to hold. (For an account of some of the model-theoretic modifications which are needed, see Kfoury [5])

3. Main Result.

The fact that finiteness of structures is inexpressible is well-known for first-order predicate logic, where it follows from an analysis of the "first-order theory of equality" -- what can be said in predicate logic using just equality, individual variables and the usual quantifiers and connectives. The result here depends on a new look at the quantifier elimination lemma on which that classical result can be based.

Notation: We use the symbols $\bigwedge$, $\bigvee$ for conjunctions, disjunctions of sets of wffs -- as in the definitions which follow.

For $k > 1$ and for V any finite set of individual variables, define

$$\rho_k \equiv (\exists x_1)\ (\exists x_2)\ \ldots\ (\exists x_k)\ \bigwedge \{\neg[x_i = x_j] \mid 1 \leq i < j \leq k\}$$

$$\rho_k\{V\} \equiv \bigvee \{\ \bigwedge\{\neg[y_i = y_j] \mid 1 \leq i < j \leq k\} \mid \{y_1, y_2, \ldots, y_k\} \subseteq V\}$$

Note that ρ_k holds iff there are at least k distinct elements in the domain as a whole; $\rho_k\{V\}$ holds iff there are at least k distinct elements just among values of variables in V.

Definition: a wff P is a p-primitive if it is equivalent to a boolean combination of [illegible] and of identities $[y=z]$ between its free variables. An n-rt F is p-primitive iff the wff $F(y_1, y_2, \ldots, y_n)$ for all individual variables $y_1, y_2 \ldots y_n$.

(Note: F is p-primitive iff $F(y_1, y_2 \ldots y_n)$ is p-primitive, for some distinct $y_1, y_2 \ldots y_n$ none of which occur in F.)

In terms of p-primitivity, the quantifier elimination result takes the following form.

Lemma: If P is p-primitive, and has k free variables, then $(\exists y)P$ is $\max\{p,k\}$-primitive.

Proof: From the hypothesis, we can assume P is in disjunctive form, with clauses which are conjunctions of wffs from $\{p_i \mid i \leq p\} \cup \{[x_i = x_j] \mid i \neq j\}$ and their negations. (The cases P ≡ true, P ≡ false are trivial; and any occurrence of wffs [z=z] can be eliminated in an obvious way.) Then the quantifier elimination proceeds as follows:

(1) $(\exists y)$ can be distributed over disjunctions, and then confined to those parts of conjunctions in which y occurs free in each conjunct, using the equivalences

$$(\exists y)\,[Q \vee R] \equiv [(\exists y)Q \vee (\exists y)R]$$

and $(\exists y)\,[Q \wedge S] \equiv [(\exists y)Q \wedge S]$, if y does not occur free in S.

Following (1), we can eliminate each occurrence of $(\exists y)$ by invoking either (2) or (3) below:

(2) if there is a wff of the form $[y=x_i]$ (or $[x_i=y]$) within the scope of a quantifier occurrence $(\exists y)$, then use

$$(\exists y)\,[[y=x_i] \wedge Q] \equiv [x_i/y]Q \quad \text{etc.}$$

(3) otherwise, $(\exists y)$ must be eliminated from a wff of the form

$$(\exists y) \bigwedge_{i=1}^{m} \neg [y=z_i]$$

consisting of a conjunction of inequalities; but such a wff can be replaced by: using the equivalence

$$(\exists y) \bigwedge_{i=1}^{m} \neg [y=z_i] \equiv [p_{m+1} \vee [p_m \wedge \neg p_m\{Z\}] \vee \ldots [p_2 \wedge \neg p_2\{Z\}]]$$

where $Z = \{z_1, z_2, \ldots, z_m\}$

The result follows, noting that $m \leq k-1$ in (3), which is the only manipulation which introduces new wffs p_i. □

Our main result is a corollary of the following theorem.

Theorem: Let P be any wff with free predicate letters chosen from $Y_1, Y_2 \ldots Y_m$ and with free individual variables chosen from finite set V of k symbols. There exists $N_{P,k}$ such that, whenever $F_1, F_2 \ldots F_m$ are p-primitive rts, $p \geq N_{P,k}$, with free individual variables chosen from V, then $[F_1/Y_1, F_2/Y_2, \ldots F_m/Y_m]P$ is p-primitive.

Proof: by induction on the length of P; the cases are as follows:

(I) for $P \equiv$ true, false, $[x_i = x_j]$ theorem trivial with $N_{p,k} = 0$.

(II) for $P \equiv \neg Q, [Q \wedge R], [Q \vee R]$ theorem follows from induction hypothesis with $N_{P,k} = N_{Q,k}$ or $\max \{N_{Q,k}, N_{R,k}\}$ (p-primitive wffs are closed under $\neg, \wedge, \vee$, clearly.)

if $P \equiv (\exists y)Q$, theorem follows from preceding Lemma and induction hypothesis, with $N_{P,k} = \max \{N_{Q,k}, k+1\}$

(III) for $P \equiv F(y_1, y_2 \ldots y_n)$, F an n-rt, we have one of cases (IV)-(VI) below:

(IV) If $F \equiv Y_i^n$ then $[F_1/Y_1 \ldots F_m/Y_m] P \equiv F_i(y_1 \ldots y_n)$ is p-primitive, from p-primitivity of F_i; $N_{p,k} = 0$.

(V) If $F \equiv [\lambda z_1 \ldots z_n . Q]$, then

$$P \equiv F(y_1, y_2 \ldots y_n) \equiv [y_1/z_1, \ldots y_n/z_n]Q \equiv \hat{Q}$$

and theorem follows from induction hypothesis, with $N_{P,k} = N_{\hat{Q},k}$, since $\hat{Q}$ is shorter than P.

(VI) is the interesting case. Suppose that $F \equiv [\mu X. G]$. then it turns out we can take $N_{P,k} = N_{Q,\, k+m}$, where $Q \equiv G(z_1, z_2 \ldots z_n)$, z_i distinct variables not in V. For suppose $F_1, F_2 \ldots F_m$ are p-primitive, with

$p \geq N_{Q,\ k+n}$ and with free variables restricted to V; define H, K, K_i by

$$H \equiv [F_1/Y_1,\ F_2/Y_2,\ \ldots,\ F_m/Y_m]\ G$$

$$K \equiv [\mu X.\ H]$$

$$K_o \equiv \lambda x_1 x_2 \ \ldots\ x_n .\ \underline{\text{false}}$$

$$K_{i+1} \equiv [K_i/X]H \equiv [K_i/X,\ F_1/Y_1\ \ldots\ F_m/Y_m]G.$$

Note that the induction hypothesis applies to any wff of the form $G(z_1, z_2 \ \ldots\ z_n)$, since such wffs are shorter than $P \equiv [\mu X.G]\ (y_1, y_2 \ \ldots\ y_n)$. Moreover the n-rt K_i is p-primitive iff every $K_i\ (z_1, z_2 \ldots\ z_n)$ is a p-primitive wff. By induction on i, each K_i has free variables restricted to V, and is p-primitive, from assumptions about F_i

(Note that $K_{i+1}(z_1, z_2 \ \ldots\ z_n) \equiv [K_i/X, F_1/Y_1\ \ldots\ F_m/Y_m]G\ (z_1, z_2 \ \ldots\ z_n)$ has up to k+m free variables, hence the requirement that $p \geq N_{Q,\ k+n}$).

But there are, up to equivalence, only a finite number

$< 2^{2^{p+(n+k)^2}}$ of p-primitive n-rts with free variables from V.

Hence, for some $k \neq \ell$

$$K_k \equiv K_\ell \ .$$

Finally, from monotonicity of G, H in X, we have

$$K_i \models K_{i+1} \models K \quad \text{for all } i.$$

Hence $K_k \equiv K_\ell \equiv K$, from definition of μ, so that K is p-primitive, and $K(y_1 \ \ldots\ y_n)$ is p-primitive, as required. □

[We are really invoking the easy theorem: if f is a monotone fn. on a

lattice with a least element $\perp$, and $f^k(\perp) = f^\ell(\perp)$ for some $k \neq \ell$, then

$$\mu f = f^k(\perp) = f^\ell(\perp) \ ;$$

the only complication here is that k, ℓ are to be independent of the lattice, which of course depends on the domain involved.] □

Corollary: Each wff with no free predicate letters or individual variables is equivalent to some finite boolean combination of sentences $\{p_i\}$

Proof: this is just the special case m=k=0 of the theorem. □

Corollary: Finiteness is mu-ineffable.

Proof: Suppose a wff P is satisfied just for those interpretations whose domain is finite (i.e. suppose that "finiteness is mu-effable"); we can eliminate free variables of P by considering its universal closure $(\forall x_0)\ (\forall x_1)\ \ldots\ (\forall x_N)P$, for suitable N, and by substituting $[\lambda x_1 x_2 \ldots x_n\ .\ \underline{\text{false}}\ ]$ for free predicate letters -- the resulting wff must still be satisfied iff the domain is finite, and must be equivalent to some finite boolean combination of $\{p_i\}$ from the above Corollary. But now let n be the maximum integer such that p_n occurs in this boolean combination, then P is satisfied by interpretations of size n iff P is satisfied by infinite interpretations. Contradiction. □

Note: we have, in effect, shown that the mu-calculus without free predicate letters is no more powerful than first-order predicate logic on its own. It follows that finiteness is not expressible even by an infinite set of wffs of the mu-calculus, since this property holds for first-order logic, as is well-known (see, for example Shoenfield [8]).

We can now see what is implied regarding the relationship with second-order predicate logic, by noting that finiteness is easily expressible if, for example, we allow universal quantification over binary relations:

$$(\forall X)\ [[\ (\forall x)\ (\forall y)\ (\forall z)\ [\ X(x,y) \wedge X(y,z) \rightarrow X(x,z)]$$

$$(\forall x)\ (\forall y)\ [\ X(x,y) \wedge X(y,x) \leftrightarrow [\ x=y]\ ] \wedge (\forall x)\ (\forall y)\ [\ X(x,y)$$

$$\vee\ X(y,x)]]\ \rightarrow (\exists x)\ (\forall y)\ X(x,y)]$$

(every linear ordering on the domain has a maximal element.)

4. The Relationship with $L\omega_1\omega$:

The language $L\omega_1\omega$ is obtained if one takes the closure of first-order predicate-logic (defined, say, by (I)-(V) above) with countably infinite conjunctions and disjunctions in addition to the usual finite operations. In this language finiteness is expressible, for example by the single infinite expressions $[\bigvee_{i=1}^{\infty} p_i]$, which is a well-formed formula of the language. $L\omega_1\omega$ can therefore express properties of structures not expressible in the mu-calculus.

On the other hand, a well-known result due to Dana Scott (see [4] ch.10) is that "well-foundedness" of binary relations X (that there are no infinite sequences $a_0, a_1, a_2, \ldots$ such that $X(a_i, a_{i+1})$ for all i) is not expressible in $L\omega_1\omega$. But this property is now expressible in the mu-calculus, by, for example

$$(\forall x)\,[\mu Y.\,[\lambda x.\,(\forall y)\,[\,X(y,x) \rightarrow Y(y)]]]\,(x)$$

Thus $L\omega_1\omega$ and the mu-calculus are incomparable in expressive power.

Finally, we should note the status of the (important) continuous mu-calculus, in which the monotonicity constraint of (VI) is strengthened to one of formal continuity.

Definition: F is formally continuous in a predicate letter X if no occurrence of X lies within a negated sub-wff $\neg P$ of F.

Every formally continuous n-ary can be seen to define a continuous functional in the lattice-theoretic sense, so that the stronger version of the fixpoint theorem, that $\mu f = \bigcup_{n=1}^{\infty} f^n(\perp)$ for any continuous function f, now holds.

But this implies that formally continuous mu-terms can be eliminated in

favour of infinite disjunctions, i.e.

$$[\mu X.F\,]\ (y_1, y_2 \ldots y_n) \equiv \bigvee_{i=0}^{\infty} G_i\ (y_1, y_2 \ldots y_n)$$

where $G_0 \equiv [\,\lambda x_1 x_2 \ldots x_n.\ \underline{\text{false}}\,]$

$$G_{i+1} \equiv [\,G_i/X\,]\ F$$

So that there is a systematic method of translating from the continuous mu-calculus into $L\omega_1\omega$.

Hence the <u>continuous</u> mu-calculus is strictly weaker in expressive power than both $L\omega_1\omega$ and the full (monotone) mu-calculus. Neither finiteness nor well-foundedness is expressible in the continuous calculus. Nevertheless the continuous mu-calculus is itself still strictly stronger than first-order logic, by the arguments presented above (among other things, by the expressibility of arbitrary equivalences between program schemas).

<u>Acknowledgements</u>: the work reported here was conceived and initiated during the author's visit to the Department of Systems and Information Sciences, Syracuse University, as Visiting Professor.

References:

1. de Bakker, J.W., de Roever W.P. A calculus for recursive program schemes; in Automatata, Languages and Programming , ed. Nivat , North -Holland (1972).

2. de Bakker, J.W., Scott, D. A theory of programs; unpublished notes. IBM, Vienna (1969).

3. Hitchcock, P., Park, D.M.R. Induction rules and termination proofs; in Nivat op. cit.

4. Keisler, J. Model theory for infinitary logic; North-Holland (1971).

5. Kfoury, D. Comparing algebraic structures up to algorithmic equivalence; in Nivat op. cit.

6. Luckham, D.C., Park, D.M.R., Paterson, M.S. On formalized computer programs ; J. Comp. Sys. Sci. 4 (1970) 220-249.

7. Park, D.M.R. Fixpoint induction and proofs of program properties; in Machine Intelligence 5, Ed. Michie, Meltzer; Edinburgh University Press (1970).

8. Shoenfield, J.R. Mathematical logic; Addison-Wesley (1967).